RobustQA Using Data Augmentation

Stanford CS224N Default Project

Rebecca Pattichis
Department of Computer Science
Stanford University
pattichi@stanford.edu
Mentor: Mandy Lu

March 20, 2021

Abstract

This project aims to explore possible improvements and extensions to the RobustQA Default baseline provided by the CS224N Winter quarter staff. Our goal is to create a domain-agnostic question answering system given DistilBERT as a pre-trained transformer model. The main method attempted in this paper is that of Task Adaptive Fine Tuning (TAPT) [1], which entails a pre-training step utilizing the Masked Language Modeling task. This method was combined with experimentation on hyperparameters (batch size, number of epochs, and learning rate) to produce the highest-achieving model. Specifically, a pre-trained MLM model with a batch size of 32 yielded an EM of 42.75 and F1 of 61.14, which are each around 2 points higher than the baseline metrics.

1 Introduction

The core motivation behind research in the field of Natural Language Processing is to understand and interpret structures within language that humans are able to understand and generalize. Therefore, a new avenue of exploration within the realm of Natural Language Processing is basic reading comprehension, also known as the question answering task. Here, models are meant to emulate a human's ability to answer a question given the right context. That is, given a question and context paragraph, can a model be trained to retrieve the answer within the paragraph?

While this project aims to develop a question answering system, it also aims to address a common problem in NLP: a failure to generalize over unseen domains [2]. Research shows that a lot of NLP structures basically fail to find structures within their training set domain that actually makes is applicable for out-of-domain sets within the same task. To address this, this project attempts to create a more robust question answering system that is better at this generalization. To do this, the baseline structure is given three in-domain datasets (Stanford Question Answering Dataset (SQuAD) [3], Natural Questions [4], and NewsQA [5]) for training, and three out-of-domain datasets that will be used for evaluation.

The initial approach this paper uses is to focus on ways the in-domain training datasets can be employed and tweaked to improve the model's efforts in generalizing its knowledge from training. The motivation behind focusing on data augmentation as the main model change revolves around finding ways to make the most of our training data. Additionally, the methods chosen are meant to inherently find different meanings of the same text and/or force the model to drop assumptions about language structure that are not actually true.

2 Related Work

The baseline provided for the default is the DistilBERT pre-trained transformer model. Bidirectional Encoder Representations from Transformers (BERT) is a model that was released by Google AI Language to allow for a pre-trained NLP model that can be used for several tasks with an additional custom layer. Specifically, BERT showed success in Masked Language Modeling (MLM) as well as Next Sentence Prediction (NSP) [6]. For further accessibility, DistilBERT was released by Hugging Face, which reduces BERT's size by 40% while retaining 97% of its linguistic understanding and being 60% faster [7].

Utilizing DistilBERT on the question answering task, an intuitive approach is to find ways to make the training set representative of the out-of-domain data. In other words, the goal is to find meaningful augmentations of the in-domain data that allow for more linguistic diversity. Task Adaptive Fine Tuning [1] has shown to be useful in adapting an already pre-trained model to adapt to a new task at hand. This method uses Masked Language Modeling (MLM) along side our actual task (in this case, question answering) to fine tune DistilBERT more effectively.

On the task of data augmentation, another method to produce more diverse training data involves creating more examples through back translation. More specifically, a given input text is translated through a "pivot" language, and then back to our source language (in our case, English) [8]. Given that our training set contains a small amount of out-of-domain data, this method could be used to create unique, but similar representations of the original input that will then allow our model to learn broader linguistic structures. However, this method might not work as desired if the encoder-decoder employed is good enough that its output is almost exact to the original input. Because of this intuition, I did not prioritize this method of data augmentation.

3 Approach

The baseline model architecture that is used throughout experimentation is DistilBERT. That being said, this paper focuses less on editing structure, and more on how changes in the dataset might provide a model more information.

3.1 Baseline: DistilBERT Model Architecture

As mentioned previously, DistilBERT is a smaller, approximated version of Google's BERT model. BERT's technology simply stacks multiple transformer encoder models. This architecture was trained on the in-domain datasets (50k samples from each) to create the baseline.

3.2 Baseline+: DistilBERT on in- and out-of-domain

Upon further inspection of the code, I noticed that there was a small amount of out-of-domain inputs that were allocated for the training and dev sets, but were not being used in the preliminary baseline. Therefore, the next improvement was to include these examples (127 examples from each of the three out-of-domain sets) in the training.

3.3 Masked LM Pre-Training w/ DistilBERT

The intuition behind this model relies on the fact that masked language modeling (MLM) is known to leverage bidirectional models that also attempts to prevent the model from learning unnecessary linguistic relationships. While DistilBERT is already pretrained on the MLM task, it is not trained using our specific data. Therefore, the following algorithm was implemented to adjust the input text provided to fit the MLM task.

3.3.1 Converting QA Data to MLM Task

Currently, the data is set up to fit the QA task. That is, given an input $x = (q, p)$ where q is the question posed and p is the context paragraph, our QA model is meant to predict where the answer is located within p.

To transform this data into input text that can be employed for the MLM task, we apply masking onto the original x, where each token has a 15% chance of being replaced with a [MASK] token. Therefore, the input for the MLM task becomes the masked version of x, and the answer that the model is training to is the unmasked original version of x.

Below is an example of the original input x provided:

Question: In what year did Tesla die?

Context: Nikola Tesla (... 10 July 1856 – 7 January 1943) was a Serbian American inventor ... best known for his contributions to the design of the modern alternating current (AC) electricity supply system.

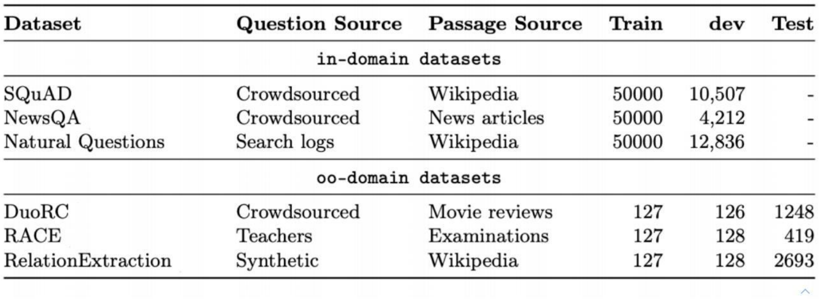
Now, using the technique of augmenting the data for the MLM task, the above example gets converted to the following:

Question: In what [MASK] did Tesla die?
Context: Nikola [MASK] (... 10 July 1856 – 7 January 1943) [MASK] a Serbian [MASK] inventor ... best known for his [MASK] to the design of the [MASK] current (AC) electricity supply [MASK].

Once this model is trained on the MLM task, the Hugging Face DistilBERT implementation for the question answering task is trained on the original data.

4 Experiments

4.1 Data



Dataset	**Question Source**	**Passage Source**	**Train**	**dev**	**Test**
		`in-domain datasets`			
SQuAD	Crowdsourced	Wikipedia	50000	10,507	-
NewsQA	Crowdsourced	News articles	50000	4,212	-
Natural Questions	Search logs	Wikipedia	50000	12,836	-
		`oo-domain datasets`			
DuoRC	Crowdsourced	Movie reviews	127	126	1248
RACE	Teachers	Examinations	127	128	419
RelationExtraction	Synthetic	Wikipedia	127	128	2693

Figure 1: Describes where the data was curated from as well as the amount of examples distributed for each set. Table borrowed from the CS224N Stanford Default Project Report, which borrowed from [9]

The data that is provided is broken up into two main chunks: the in-domain and out-of-domain sets. As mentioned previously, the in-domain data consists of a combination of SQuAD [3], NewsQA [5], and Natural Questions [4]. The source of the questions were mainly crowdsourced while the context paragraphs were mainly scraped from Wikipedia and various news articles. The out-of-domain datasets consist of DuoRC [10], RACE [11], and RelationExtraction [12].

4.2 Evaluation method

The common evaluation metrics that are used for question answering tasks are a combination of the Exact Match (EM) and F1 score. EM is simply a binary output that indicates whether or not the model's predicted answer matches the ground truth exactly or not. To give a bit more leniency when evaluating our model, we are given three different ground truth examples for the validation and test runs, so that model's predicted output will receive an EM of 1 if it exactly matches any of those.

The other metric, the F1 score, is meant to balance out the harshness of the EM metric by giving us a combined metric for precision and recall. Recall that precision measures the amount of our answer that is a subset of the ground truth, and recall measures how much our predicted outcome contained the answer:

$$F1 = \frac{2 \cdot precision \cdot recall}{precision + recall}$$

Like the EM score, we will calculate the F1 on all three ground truth provided and take the maximum score to evaluate the success of our model.

4.3 Experimental details

The baseline model provided by the CS224N team consisted of training with a batch size of 16, learning rate of 3e-5, and 3 epochs on the in-domain training set, which took about 14 hours to train. The immediate improvement made on this model was Baseline+, which simply added a little over 300 input examples (from the out-of-domain set) into the training set. The same default settings were used as the baseline, and took the same amount of time to train.

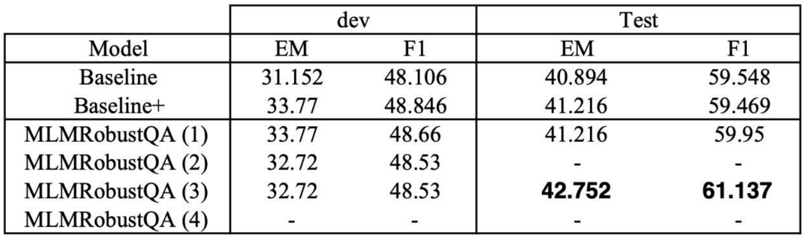
For upcoming models, I decided to first perform masked language modeling using the in- and out-of-domain sets before training for the question answering task. MLM also employed the default hyper parameters listed above, and took about 12 hours to train. From here, I experimented by adjusting different hyper parameters.

Model	Batch Size	Learning Rate	Epochs
Baseline	16	3e-5	3
Baseline+	16	3e-5	3
MLMRobustQA (1)	16	3e-5	3
MLMRobustQA (2)	32	3e-5	3
MLMRobustQA (3)	32	3e-5	6
MLMRobustQA (4)	32	3e-4	4

Table 1: Table outlining the experimental details of each model trained.

4.4 Results

Figure 2 gives evaluation metrics on the dev set for each model that was trained. Based on these values, the top models were evaluated on the test set. Note that MLMRobustQA (4) does not have results because its training loss was 2x larger than the baseline and other models, so it was stopped prematurely to avoid wasting Azure credits. The lack of convergence is very likely due to the alteration in learning rate, where I decided to follow my intuition that a greater



	dev		Test	
Model	EM	F1	EM	F1
Baseline	31.152	48.106	40.894	59.548
Baseline+	33.77	48.846	41.216	59.469
MLMRobustQA (1)	33.77	48.66	41.216	59.95
MLMRobustQA (2)	32.72	48.53	-	-
MLMRobustQA (3)	32.72	48.53	**42.752**	**61.137**
MLMRobustQA (4)	-	-	-	-

Figure 2: Evaluation metrics on the models

batch size might allow for a larger learning rate. However, this experiment did not follow through: where the initial training loss for the models in Figure 2 are at around 3, MLMRobustQA (4) stayed consistently at around 6 without dropping.

Analyzing Figure 3, which represents the training loss, we see that there are basically two different categories of loss: the one surrounding the baseline and the one below. The models that surround the baseline (orange) include Baseline+ and MLMRobustQA (1). It makes sense that Baseline+ is so similar in training loss; after all, we only included 381 new samples from the out-of-domain data, which is trivial compared to the original training dataset size. What is important to notice is that MLMRobustQA (1) is also around the baseline training loss, which signifies to me that the pre-training with MLM using DistilBERT was not too effective at constructing a better QA model.

It turns out that increasing the batch size from 16 to 32 and keeping the current learning rate of 3e-5 is what helped the training loss decrease. Note that MLMRobustQA (2) and (3) are basically the same, (3) just runs for twice as many epochs. This means that ultimately, experimenting with hyperparameters provided more effective and productive changes to the baseline over the MLM pre-training.

Through these results, we can say that doing MLM pre-training using DistilBERT was not sufficient enough to improve the baseline. In retrospect, since DistilBERT is already trained with the goal of optimizing MLM tasks, it makes sense that simply adding a few more examples to learn from does not greatly affect the weights in the model.

5 Analysis

When reviewing the baseline text predictions, it seems like the thing the model is trying to learn the most are the indices of the answer. In other words, during the earlier steps of evaluation, if the baseline answered the question properly, it was usually overshot with extra padding around the actual answer (this means that through training, F1 is being improved on by reducing this padding). Interestingly enough, MLMRobustQA (3) seems to have the opposite problem: its

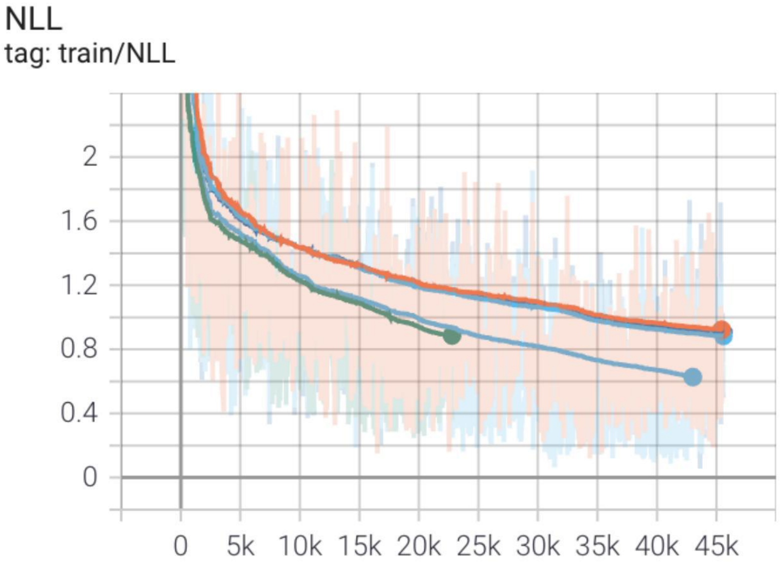


Figure 3: Graph of the training loss for the following models: Baseline (orange), Baseline+ (dark blue), MLMRobustQA (1) (light blue), MLMRobustQA (2) (dark green), and MLMRobustQA (3) (light blue by the dark green)

prediction is undershooting the length of the actual answer. Additionally, this model does not do well on contexts that are relatively small compared to the others in the data, which intuitively makes sense. Below are time step examples as comparison on their learning patterns.

Baseline: Step 0 vs. Step 40

- Question: What is the name of the hospital where Gary Coleman was admitted?
- Shortened Context: Actor Gary Coleman is in critical condition in a Provo, Utah, hospital, a hospital spokeswoman said Thursday.... spokeswoman for Utah Valley Regional Medical Center, confirmed that Coleman, 42, was being treated there after being admitted on Wednesday...
- Answer: Utah Valley Regional Medical Center,
- Prediction: for Utah Valley Regional Medical Center, confirmed that Coleman, 42, was being
- Question: What sort of system releases the exhaust steam into the atmosphere?

- Shortened Context: The working fluid in a Rankine cycle can operate as a closed loop system, where the working fluid is recycled continuously, or may be an "open loop" system....
- Answer: open loop
- Prediction: open loop" system

MLMRobustQA (3)

- Question: who is the minister of youth in namibia
- Context: BLiB Minister of Sport , Youth and National Service : Jerry Ekandjo (until February 2018) , Erastus Utoni BUlB BLiB Deputy : Agnes Tjongarero EELiE EEUlE EELiE
- Answer: Erastus Utoni
- Prediction: National Service : Jerry Ekandjo (until February 2018) , Erastus Utoni BUlB
- Question: Where are pyrenoids found?
- Shortened Context: The chloroplasts of some hornworts and algae contain structures called pyrenoids. They are not found in higher plants....
- Answer: The chloroplasts of some hornworts and algae
- Prediction: hornworts and algae

6 Conclusion

Overall, the best performing model was MLMRobustQA (3), which employed pre-training MLM using our dataset and DistilBERT on the given dataset, as well as a batch size of 32 during training for the question answering task. These experiments suggest that tuning hyperparameters is an avenue worth investigating for improving a domain-agnostic QA system. Despite it not being the main reason for improvement, I implemented data augmentation for masked language modeling and was able to apply newly acquired NLP knowledge on an innovative challenge within this realm of research.

Note that the biggest limitation with this project is that multiple forms of data augmentation were not extensively tested, so it cannot be determined how effective and/or ineffective these methods may be compared to tuning hyperparameters. In the future, a combination of data augmentation to create a larger corpus of input data would be the route suggested.

References

[1] Suchin Gururangan, Ana Marasovic, Swabha Swayamdipta, Kyle Lo, Iz Beltagy, Doug Downeey, and Noah A Smith. Don't stop pretraining: Adapt language models to domains and tasks. In *arXiv preprint arXiv:2004.10964*, 2020.

[2] Robin Jia and Percy Liang. Adversarial examples for evaluating reading comprehension systems. In *arXiv preprint arXiv:1707.07328*, 2017.

[3] Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. Squad: 100,000+ questions for machine comprhension of text. In *CoRR, abs/1606.05250*, 2016.

[4] Tom Kwiatkowski, Jennimaria Palomaki, Olivia Redfield, Michael Collins, Ankur Parikh, Chris Albeerti, Danielle Epstein, Illia Polosukhin, Matthew Kelcey, Jacob Devlin, Quoc Le, and Slav Petrov. Natural questions: a benchmark for qustion answering research. In *Association for Computational Linguistics (ACL)*, 2019.

[5] Adam Trischler, Tong Wang, Xingdi Yuan, Justin Harris, Alessandro Sordoni, Philip Bachman, and Kaheer Suleman. Newsqa: A machine comprehension dataset. In *ACL 2017*, 2017.

[6] Jacob Devlin, Ming-Wei Chang, Kenton Le, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformres for language understanding. In *arXiv preprint arXiv:1810.04805*, 2018.

[7] Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter. In *arXiv preprint arXiv:1910.01108*, 2019.

[8] Shayne Longpre, Yi Lu, Zhucheng Tu, and Chris DuBois. An exploration of data augmentation and sampling techniques for domain-agnostic question answering. In *arXiv preprint arXiv:1912.02145*, 2019.

[9] Adam Fisch, Alon Talmor, Robin Jia, Minjoon Seo, Eunsol Choi, and Danqi Chen. Mrqa 2019 shared task: Evaluating generalization in reading comprehension. In *Workshop on Machine Reading for Question Naswering (MRQA)*, 2019.

[10] Ammrita Saha, Rahul Aralikatte, Mitesh M. Khapra, and Karthik Sankaranarayanan. Duorc: Towards complex language understanding with paraphrased reading comprehension. In *ACL*, 2018.

[11] Guokun Lai, Qizhe Xie, Hanxiao Liu, Timing Yang, and Eduard Hovy. Race: Large-scale reading comprehension dataset from examinations. In *EMNLP*, 2017.

[12] Omer Levy, Minjoon Seo, Eunsol Choi, and Luke Zettlemoyer. Zero-shot relation extraction via reading comprehension. In *arXiv preprint arXiv:1706.04115*, 2017.

A Appendix

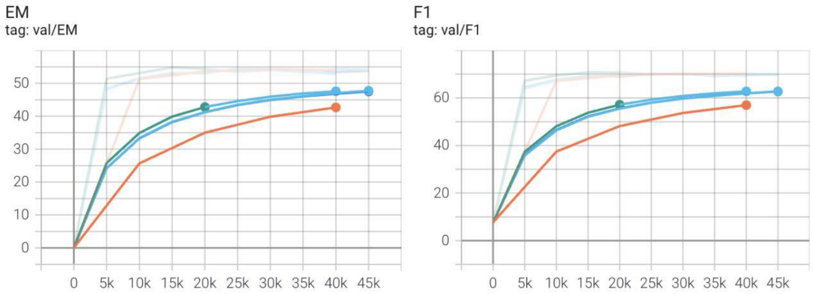


Figure 4: Graph of the dev EM and F1 scores for the following models: Baseline (orange), Baseline+ (dark blue), MLMRobustQA (1) (light blue), MLMRobustQA (2) (dark green), and MLMRobustQA (3) (light blue by thee dark green)